CONDITIONAL LEAVING-ONE-OUT AND CROSS-VALIDATION FOR DISCOUNT ESTIMATION IN KNESER-NEY-LIKE EXTENSIONS

J. Andrés-Ferrer[*]

Pattern Recognition and
Human Language Technology
Universidad Politécnica de Valencia

M. Sundermeyer[†] *and H. Ney*

Human Language Technology
and Pattern Recognition
RWTH Aachen University

ABSTRACT

The smoothing of n-gram models is a core technique in language modelling (LM). Modified Kneser-Ney (mKN) ranges among one of the best smoothing techniques. This technique discounts a fixed quantity from the observed counts in order to approximate the Turing-Good (TG) counts. Despite the TG counts optimise the leaving-one-out (L1O) criterion, the discounting parameters introduced in mKN do not. Moreover, the approximation to the TG counts for large counts is heavily simplified. In this work, both ideas are addressed: the estimation of the discounting parameters by L1O and better functional forms to approximate larger TG counts. The L1O performance is compared with cross-validation (CV) and mKN baseline in two large vocabulary tasks.

Index Terms— Leaving-One-Out, Language Modelling, Cross Validation, modified Kneser-Ney smoothing

1. INTRODUCTION

Language modelling (LM) consists in estimating a probability distribution for arbitrary word sequences. A LM is expected to distribute high probability to correct sentences while giving reasonable probabilities to unseen or unlikely sentences. This is, however, a challenging task since words occurring in a corpus are dominated by the singletons. Hence, a competitive LM must calculate probability estimates for unseen or infrequent events, the so called *small probabilities* [1, 2].

Several techniques and attempts have emerged for LM, although few of them are competitive. Recent efforts include maximum entropy (MaxEnt) [3] and neural networks [4]. However, one of the most widespread techniques is the n-gram models [5]. Despite their simplicity, n-gram models are fast and competitive in terms of both perplexity (PPL) and word-error rate (WER). Experimentally, the n-grams performance is mainly achieved by how they estimate small probabilities, the so-called *smoothing techniques* [5]. In contrast, from a theoretical point of view some authors claim that n-gram models approximate MaxEnt models [6].

The Kneser-Ney (KN) smoothing technique stands out because of its performance [5]. This technique approximates the Turing-Good (TG) counts [1], by discounting a fixed parameter b to observed counts. This parameter, like TG counts, is estimated by *leaving-one-out* (L1O) [1, 2]. The discounted probability is redistributed among all events according to a *generalised smoothing distribution* [7].

Several techniques derive from the initial absolute discounting KN smoothing. On the one hand, the standard KN discounting approximates all the TG counts by means of one single discounting parameter [2]. On the other hand, the modified KN (mKN) discounting [5] uses 3 parameters b_1,b_2,b_{3+}; for discounting singletons, doubletons and larger n-gram counts, respectively. Although the mKN parameters are heuristically estimated by *Chen's approximation* [5], they report better performance than the KN smoothing [5]. PPLs are slightly improved by extending the mKN method to a larger number of discounting parameters [8], despite they are estimated optimising the *joint L1O PPL*. Recently, an alternative estimation by cross-validation (CV) [9] for the former extension incurred further improvements. However, it is yet to investigate the estimation of several discounting parameters with the foundation of the smoothing method, namely, the (conditional) L1O criterion. On the one hand, this is due to a lack of closed form solutions; and on the other hand, a global optimisation by numerical methods would incur unfeasible training times. In this paper, we optimise the conditional L1O PPL locally for each n-gram level obtaining improvements of 4% of PPL. Furthermore, we compare the L1O with the recent CV scheme in [9], showing that both obtain similar results. The mKN baseline is also extended to estimate more than 3 discounts by generalising Chen's heuristic formula [5].

Finally, the large counts are widely ignored in the discounting scheme since they are usually discounted by a fixed value. In this work, we extend mKN smoothing with different discounting schemes for the larger counts, namely: a *logarith-*

[*]Work supported by the Spanish MEC/MICINN under iTrans2 project (TIN2009-14511), and by the Spanish MITyC under the erudito.com (TSI-020110-2009-439) project; as well as by the Generalitat Valenciana grant Prometeo/2009/014 and GV/2010/067, and by grant UPV/2009/2851.

[†]This work was partly realized as part of the Quaero programme, funded by OSEO, French State agency for innovation.

mic absolute discounting and a *linear* absolute discounting. These schemes are inspired by the MaxEnt models [6].

2. DISCOUNTING MODEL

Given an n-gram hw, where h is the context of words preceding the observed one w; the absolute discounting scheme proposed by the KN technique subtracts a fixed quantity from the observed counts $N(h,w)$ in order to gain a probability mass $\gamma(h)$ that is redistributed accordingly to a generalised discounting distribution $p(w \mid \bar{h})$; where $\bar{h}$ denotes a generalised context obtained by dropping the leftmost context word. The generalised discounting distribution is recursively smoothed using the same scheme but with different counts [7]. This model is referred to as interpolated KN smoothing.

In this paper, we propose a twofold extension to the model. On the one hand, we use B different discounts for the B smaller counts [8]. On the other hand, we add a coefficient parameter c_+ and a *discounting function*, $g(r)$, for large counts. The proposed model is expressed as follows

$$p(w \mid h) = q(w \mid h) + \gamma(h) p(w \mid \bar{h}) \tag{1}$$

where $\gamma(h)$ is the gained probability mass, and where $q(w \mid h)$ is the discounted "probability[1]" for the n-gram hw, i. e.

$$q(w\,|\,h) = \begin{cases} \frac{N(h,w) - b_B - c_+ g(N(h,w))}{N(h)} & N(h,w) \geq B \\ \frac{N(h,w) - b_{N(h,w)}}{N(h)} & 0 < N(h,w) < B \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

with the parameters $\{b_1^B, c_+\}$ for each distribution in the smoothing hierarchy $(n, n-1, \ldots, 1)$. The gained probability mass is

$$\gamma(h) = \frac{\theta(h)}{N(h)} = \frac{\sum_{r=1}^{B-1} b_r n_r(h) + b_B n_+(h) + c_+ n_g(h)}{N(h)} \tag{3}$$

where $\{n_r(h)\}_{r=1}^{B-1}$ stands for the so-called *count-of-counts (CoC)*, i.e. the number of n-grams that have occurred exactly r times preceded by the context h; where $n_+(h)$ is the CoC for the large counts, $n_+(h) = \sum_{r \geq B} n_r(h)$; and, finally, $n_g(h)$ is defined as $n_g(h) = \sum_{r \geq B} n_r(h) g(r)$ by analogy.

In this work, 3 different discounting functions are analysed: the logarithm, $g(r) = \log(r)$; a linear function, $g(r) = r$; and the standard function $g(r) = 0$. The latter case is equivalent to an extension of mKN smoothing [8, 9] for $B = 3$.

3. DISCOUNTING PARAMETER ESTIMATION

Given the model defined in the previous section, three estimation methods are considered: leaving-one-out (L1O), cross validation (CV) and Chen's approximation.

[1]Since counts are discounted, it is actually not a probability by itself.

3.1. Chen's approximation

Although Chen's approximation was introduced [5] to estimate the 3 parameters of the mKN smoothing; it is easily extensible to more discounts as follows:

$$b_r = r - (r+1) b \frac{n_{r+1}}{n_r}, \tag{4}$$

where b denotes the Kneser-Ney approximation given by $b = n_1/(n_1 + 2n_2)$ with n_1, n_2 being the CoC for singletons and doubletons, respectively. The parameter c_+ is fixed to 0 in this case, which corresponds to the standard function.

3.2. Cross Validation (CV)

In this method [9], a held-out set approximates the test PPL, and, then, we maximise its log-likelihood criterion defined as

$$F_{CV} = \sum_{hw} C(h,w) \log p(w|h) \tag{5}$$

where $C(h,w)$ denotes the held-out count of the n-gram hw.

Similarly to [9], we optimise F_{CV} by means of the improved resilient Back-Propagation (Rprop) [10], which requires to compute the gradient of F_{CV}. For a given count r, the gradient for b_r is the same as that presented in [9]. As for the new additional parameter c_+ the gradient is given by

$$\frac{\partial F_{CV}}{\partial c_+} = \sum_{hw} C(h,w) \frac{1}{p(w|h)} \frac{\partial}{\partial c_+} p(w|h) \tag{6}$$

and subsequently for the higher n-gram order level

$$\frac{\partial}{\partial c_+} p(w|h) = \frac{n_g(h)}{N(h)} p(w|\bar{h}) - \begin{cases} \frac{g(N(h,w))}{N(h)} & N(h,w) \geq B \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

For lower orders, the gradient is premultiplied by the discounted mass from the higher orders, similarly to [9].

3.3. Leaving-one-out (L1O)

Similarly to standard KN smoothing, in this method, the discounting parameters are set to maximise the (conditional) L1O log-likelihood (or equivalently minimise the L1O PPL), which is known to be a reliable estimate of the test PPL [2]

$$F_1 = \sum_{hw} N(h,w) \log p_1(w|h) \tag{8}$$

where $p_1(w|h)$ stands for the L1O probability, which is obtained by leaving each n-gram occurrence hw out for testing and training a model with the remaining data. For the model proposed in section 2, the L1O probability is

$$p_1(w \mid h) = q_1(w \mid h) + \gamma_1^{N(h,w)}(h) p(w \mid \bar{h}) \tag{9}$$

where

$$\gamma_1^r(h) = \begin{cases} \frac{\theta(h)}{N(h)-1} & r > B \\ \frac{\theta(h) - b_B - c_+ g(B) + b_{B-1}}{N(h)-1} & r = B \\ \frac{\theta(h) - b_r + b_{r-1}}{N(h)-1} & \text{otherwise} \end{cases} \tag{10}$$

Corpus	Set	Words	OOV	Domain
Quaero EN	train	348.0M	1.28%	blog+forum
	held-out	41.8K	0.45%	transcriptions
	test	1.2M	0.49%	transcriptions
Quaero FR	train	243.4M	1.15%	blog+forum
	held-out	46.7K	0.45%	transcriptions
	test	700.4K	0.01%	transcriptions

Table 1. Statistics for the corpora splits, including Out-of-vocabulary rates (OOV). Vocabulary sizes are 150K, and 200K for Quaero EN(glish) and FR(ench), respectively.

with $b_0 = 0$, and where for the "otherwise" cases, if $N(h) = 1$, then $0/0$ is fixed[2] to b_1. Finally $q_1(w \mid h)$ is equal to

$$q_1(w|h)=\begin{cases}\frac{N(h,w)-1-b_B-c_+g(N(h,w)-1)}{N(h)-1} & N(h,w)>B \\ \frac{N(h,w)-1-b_{N(h,w)-1}}{N(h)-1} & \text{otherwise}\end{cases} \quad (11)$$

Although, the L1O criterion is eligible for a global optimisation of the parameters for all orders at once, the required computational resources render this approach unfeasible. Hence, we optimise Eq. (8) for each order independently, in n local optimisation rounds, like standard KN smoothing. This speeds up the training and each local optimisation is convex. Since, there is not known closed-form solution, the Rprop is used like CV [9].

In this case, the gradient of F_1 in Eq. (8) is given by

$$\frac{\partial F_1}{\partial b_r}=\sum_{hw}\frac{N(h,w)}{p_1(w|h)}[p(w|\bar{h})\frac{\partial\gamma^{N(h,w)}(h)}{\partial b_r}-\frac{\partial}{\partial b_r}q_1(w|h)] \quad (12)$$

where we omit the partial derivatives of $\gamma^r(h)$ and $q_1(w|h)$ for simplicity sake and space constraints. However, they are easily obtained from Eqs. (10) and (11).

4. EXPERIMENTS

In order to compare the proposed estimation methods as well as the different functional schemes, we ran several experiments on two different large vocabulary corpora, namely, Quaero-EN [3], and Quaero-FR. A summary of some statistics about the corpora are reported in Table 1.

For our experiments, we did not use LM interpolation to avoid masking the discount optimisation effects. We used 4-grams in oder to report comparable results to [9].

For assessing the performance of the 3 proposed estimation techniques, we used the PPL in the test set for both corpora. Furthermore, 3 different discounting functions were analysed for discounting large counts: the standard function; a logarithmic function; and, a linear function. Finally, different number of discounting parameters were analysed, ranging from the standard mKN (3 discounts) to 40. This experimentation is reported in Table 2.

[2]Actually, we take a fraction $\varepsilon < 1$ of unique event out to test while the remaining fraction is kept for training; and we take the limit $\varepsilon \to 1$.

[3]Quaero research programme, see http://quaero.org.

Model	mKN	Chen's Approx.	L1O	CV
Quaero-EN	20.7	20.7	**20.5**	**20.5**

Table 3. Word Error Rates for Quaero-EN (LM rescoring)

Several conclusions are drawn from Table 2. Firstly, CV works slightly better than L1O. This supports the conclusion that CV does not over-train unless an large number of discounts is used [9]. The discrepancy between CV and L1O is due to the different optimisation criterion used for each method. In CV, a global optimiation for all parameters is peform to minimise the highest order CV PPL, whereas L1O uses n local optimisation that minimise the L1O PPL for earch order. Secondly, although most of the performance is obtained using 10 discounts, the linear discounting function seems a better model for discounting larger counts. Finally, it is surprising that the PPL is improved by simply increasing the number of discounting parameters, which are estimated with Chen's approximation. When compared with mKN (3 discounts), increasing the parameters to 40 improves the PPL up to 2.3% for Quaero-EN . This improvement is larger if we use L1O (4.4%) and even larger for CV (5.5%).

In order to assess whether the PPL improvements imply better systems, we have computed WER for Quaero-En. We have created lattices for the test data using the state-of-the-art acoustic models of the single best system described in [11] and a standard mKN LM. Then, we applied an LM re-scoring step using models with 40 discounts and a linear discounting function estimated with the three methods: Chen's approximation, L1O and CV. Table 3 reports recognition results on the Quaero evaluation corpus 2010 [11]. Significance tests were not reported because of test size (41.0K words). Note that the Chen's approach improves the PPL but not the WER. Concerning to L1O and CV, both obtain the very same result, improving the WER in 0.2 points.

5. DISCUSSION

Independent of the optimisation criterion, a reasonable estimation of the discounting parameters yields most of the improvements. Both methods, CV and L1O, obtain the same performance in terms of WER; although in terms of PPL, L1O is slightly worse than CV even though L1O is convex. This is probably due to the fact that we are optimising the L1O PPL for each order level independently and not in one single global optimisation as CV. This local optimisation is less exact when estimating the test PPL. A global L1O optimisation at a highest order, like CV, would better approximate the test perplexity. However, it will require unfeasible computational resources. Moreover, a subsampling of the training data could speed up the method, but then the subsampling would

Quaero-EN	Chen's approx.	Leaving-one-out			Cross Validation		
Num. of discounts \ Disc. function	standard	standard	logarithmic	linear	standard	logarithmic	linear
3	209.3	202.4	200.3	200.2	200.7	198.1	197.4
10	206.3	200.7	200.2	**199.8**	198.4	197.9	**197.1**
20	205.3	200.3	200.1	**199.8**	198.0	197.7	197.2
40	**204.4**	**200.1**	**200.0**	**199.8**	**197.8**	**197.7**	197.3
Quaero-FR	Chen's approx.	Leaving-one-out			Cross Validation		
Num. of discounts \ Disc. function	standard	standard	logarithmic	linear	standard	logarithmic	linear
3	183.2	179.4	178.3	178.5	177.4	175.9	175.8
10	180.9	178.2	**178.2**	178.3	175.5	175.2	**175.0**
20	180.0	**178.0**	**178.2**	**178.2**	175.3	**175.1**	**175.0**
40	**179.4**	**178.0**	**178.2**	**178.2**	**175.2**	**175.1**	**175.0**

Table 2. Perplexities in the test set for several estimation techniques and for 3 discounting functions: standard extended mKN smoothing ($g(r) = 0$), logarithmic discounting ($g(r) = \log(r)$); and linear discounting ($g(r) = r$). The number of discounting parameters (B) are 3, 10, 20 and 40.

also affect the training counts during the optimisation process rendering this method unsatisfactory when compared to the CV, that does not modify the training counts. Note that the time required for computing the gradient in the CV case depends on the number of n-grams that occur in the held-out set, whereas the for the L1O it depends on the n-grams that occur on the training data. Therefore, the CV is much faster than the L1O. Specifically, the CV does not incur a significant delay with respect to the standard smoothing techniques as long as the size of the held-out set is small.

The extension of the Chen's approximation does not obtain WER improvements although it reports PPL improvements. From our point of view, this is surprising since this method does not accurately approximate the test PPL, but makes rough approximations instead. Our conclusion is that the test PPL, despite being non-convex, is smooth and flat as a function of the discounting parameters except for extreme values such as 0.

Regarding the functional form of the discounting for large counts, the proposed functions do not better approximate the TG counts since improvements in terms of PPL are negligible.

In summary, the KN discounting family introduces two main advantages: the absolute discounting scheme and the generalised smoothing distribution. From this work, it is concluded that only small improvements are obtained extending the discounting scheme, although a more extensive experimentation might be necessary. In [5], it is claimed that the superiority of the mKN/KN discount is mainly due to the generalised smoothing distribution, and consequently we think that if there is room for improvement for this family of discounting techniques, it is on the generalised discounting distribution, and not in the discounting parameters, at least, ignoring the generalised distribution, no matter whether this optimisation is based on CV or on L1O.

6. REFERENCES

[1] A. Nadas, "On Turing's formula for word probabilities," *IEEE Trans. on Acoustics, Speech and Signal Proc.*, vol. 33, no. 6, pp. 1414–1416, 1985.

[2] H. Ney, "On the estimation of small probabilities by leaving-one-out," *IEEE Trans. on Patt. Analysis and Machine Int.*, vol. 17, no. 12, pp. 1202–1212, 1995.

[3] S. Chen, "Shrinking exponential language models," *Proceedings of Human Language Technologies*, 2009.

[4] H. Schwenk, "Continuous space language models," *Computer Speech and Language*, 2007.

[5] S. Chen, "An empirical study of smoothing techniques for language modeling," in *Proceedings of ACL*, 1996.

[6] J. Goodman, "Exponential priors for maximum entropy models," *Proc HLT-NAACL*, 2004.

[7] R. Kneser and H. Ney, "Improved backing-off for M-gram language modeling," in *ICASSP*, 1995.

[8] J. Andrés-Ferrer and H. Ney, "Extensions of absolute discounting (Kneser-Ney method)," in *ICASSP 2009*.

[9] M. Sundermeyer, R. Schlüter, and H. Ney, "On the estimation of discount parameters for language model smoothing," in *Interspeech*, 2011.

[10] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: the RPROP algorithm," in *IEEE Int. Conf. on Neural Networks*, 1993.

[11] M. Sundermeyer et al., "The RWTH 2010 Quaero ASR Evaluation System for English, French, and German," in *ICASSP 2011*.